ZigBee SE 1.0 security analysis

Introduction

This document is to assist the CSWG standards subgroup with respect to the review of SE 1.0 security.

The issues outlined here do not infer that the SE 1.0 specification is insecure. Like the Carnegie-Mellon report, it aims to highlight certain issues which need to be considered from an implementation point of view and also from a standardization point of view with regard to potential future revisions.

Note these comments will apply to the equivalent sections in SE 1.1. There is fundamentally no difference in the security mechanisms between SE 1.0 and SE 1.1. Some clarification has been done in SE 1.1 with regard to some of the implementation issues.

This document only looks at the sections directly relevant to cyber security.

Author

Robert Cragie
Gridmerge Ltd.
89 Greenfield Crescent
Grange Moor
Wakefield
WF4 4WA
United Kingdom

Tel. +1 415 513 0064 (US)
Tel. +44 1924 910888 (UK)

robert.cragie@gridmerge.com

Glossary

CBKE: Certificate-based Key Establishment

ESI: Energy Services Interface (aka ESP: Energy Services Portal in SE 1.0)

KEC: Key Establishment Cluster: The cluster used to perform key establishment

PKI: Public Key Infrastructure

TC: Trust Center

Summary of issues

Permit join flag

The permit join flag is used to indicate via the 802.15.4 beacon that it is possible to join the network. This is generally configured in the ZigBee Coordinator and propagated throughout the ZigBee

Routers in the network so they can also advertise in their beacons for potential joining devices. The device will then be able to initiate joining with the ZigBee Coordinator or the nearest ZigBee Router.

The aim is to turn this on over the backhaul at the ZigBee Coordinator (usually the meter/ESI) in conjunction with registration information for one or more devices (identity and installation code), which will also initiate the propagation. However, due to the general difficulty of sending over the backhaul, the permit join flag is often left on and registration information is not sent to the ESI/TC. This can cause issues where devices steer to neighboring networks and also makes honeypot rogue devices more likely. Mitigation is through proper use of registration and joining mechanisms.

Installation code

The installation code is a number (supposed to be random) associated with the manufacture of a device, which is used to secure transport of the network key. However, the installation code is not further randomized (salted) in any way. It is passed through a cryptographic hash to provide a TC link key, which is then used through a predictable HMAC (i.e. HMAC of key concatenated with 0x00) to encrypt the key. Thus, using a weak installation code, an attacker can send an association request and a network key request, which will be sent encrypted using a weak installation code. The attacker can then simply try all combinations of installation code to decrypt the message until the MIC correlates. The attacker then has the network key.

SE 1.1 mitigates by insisting installation codes are at least 64 bits in length.

Another possible solution is to salt the key somehow. This is difficult though, given the limited choices of salting information available to a legitimate joiner and the ESI. The proper fix is to use the public key establishment to create a pairwise key which would be used to deliver the key as is done in SE 2.0, and as suggested in the CMU audit.

Overlapping link key domains

The key used to authenticate for network access is the same key used for application layer transport, therefore the key domains overlap. Ideally, a derivation of the key would be used. This key also 'replaces' the original installation code, which as mentioned is used as the basis for a link key. This causes various issues in the specification (see detailed analysis).

163-bit key sizes

These are equivalent to 80-bit symmetric keys. NIST SP 800-57 and NISP SP 800-131A list lifetimes of keys and suggests that these are already deprecated. To some extent it depends on the application as to whether this is a significant problem but it is likely there will be classes of devices which would need higher key strengths.

Inter-PAN mechanism

Inter-PAN mechanism is totally unsecure and should not be used. It is not currently specified for any communication in SE 1.0 as is really a placeholder. However, this does not stop it potentially being used in the future.

Certificate weakness

The certificate specification only mentions profile specific information; it does not explicitly cover any device type or security level thus making it possible to masquerade as a different device which

may have more external security measures, e.g. a IHD (low cost, no hardened security protection) could masquerade as a meter.

ZigBee PRO issues

There are certain issues in the underlying ZigBee PRO specification which also need to be considered.

- No MAC layer security
- No use of challenge/response to synchronize frame counter to mitigate against replay attack (i.e Entity Authentication)
- Broadcast of new network key does not mitigate against insider attack
- Rejoin has implicit authorization with regard to access control
- No explicit Trust Center policies

No MAC layer security

The implication of having no security protection on the MAC layer frames is that the MAC header (MHDR) part of the MAC PDU is not protected. The MAC payload is protected by means of the NWK layer security protection. This means that the MHDR fields have no integrity check on them and can thus be intercepted and altered.

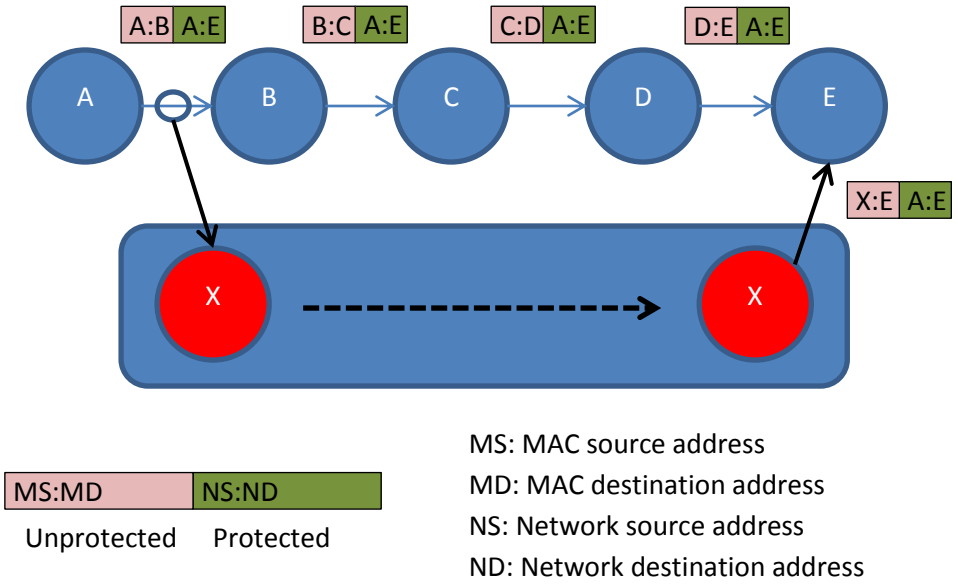

Figure 1: Wormhole replay attack

This can be exploited in wormhole replay attack. This is quite a sophisticated attack and requires knowledge of the network topology.

The scenario is where node A wants to send a packet to node E. Routing is already established to node E.

1. On the first hop, the MAC packet is sent from A to B, with network source A and the network destination E. The network addresses are protected by the MIC at the end of the MAC payload but the MAC addresses are not protected.

2. The rogue node X overhears and records the MAC packet sent from A to B.
3. The rogue device X has the ability to tunnel the overheard packet A:B|A:E to a node in radio range of the network destination E. It modifies the MAC source address to be itself (or any other node) and the MAC destination address to be E and transmits the packet as X:E|A:E. Note node X needs no knowledge of the network key or any other key
4. Node E receives the packet from X. It has no knowledge of X but the MAC payload, as it has been unmodified, passes security processing. The fact the MAC header has been modified cannot be detected by E. It therefore assumes that this is the first correct message from X and updates its internal records for X and passes it on for further processing
5. Sometime later, the valid message from D:E|A:E will be received by E containing the same payload as the rogue packet. It will also pass security processing and thus be sent for further processing and is thus processed erroneously twice.

The way of mitigating against this attack is to only accept packets from authenticated neighbors, using a challenge/response based on the network key. This would mean E would not process the packet from X as it is not an authenticated neighbor, and would require any rogue device to know the network key. The Entity Authentication mechanism in the ZigBee PRO specification was designed to provide neighbor authentication and to synchronize frame counters between neighbors but it is not used in SE 1.0

Broadcast of new network key

One reason for wishing to rotate the network key is to limit its validity period and thus the ability for a compromised node to be able to decrypt any volume of traffic. For example, this was the temporary fix (TKIP) introduced for 802.11 when it was realized the WEP encryption was very weak and the keys could be brute force cracked easily.

Key updates in ZigBee PRO are accomplished by broadcasting the new network key to all devices secured with the old network key. Therefore, any compromised device will be able to receive the new network key. It should be possible to unicast the new network key to all devices except the known compromised device then switch to the new network key. The key encryption key (KEK) would be derived from the TC link key instead of old network key. In this way, a compromised device can be orphaned. This of course assumes knowledge of which device in particular is compromised

Detailed analysis of 075356r15 (ZigBee SE 1.0 specification)

5.1 pg 9 ln 24-28: Specifies use of application layer security and peer-to-peer link keys.

5.1 pg 9 ln 31-36: Public pricing mechanism – allow information without joining. This is insecure and should not be used as a receiving device cannot verify the authenticity of the information or its source.

5.2 pg 11 ln 39 – pg 12 ln 18. This section provides more requirement constraints on the ZigBee and ZigBee PRO stack profile with regard to features which are optional in the stack profiles.

5.3.1 pg 13 ln 34-36: No master key – does not allow key establishment using shared secret.

5.3.1 pg 14 ln 7-9: It should not be possible to pre-install the network key. This would imply some mechanism whereby locally the network key can be extracted from another device on the network

through a means other than joining. If this is to take place, this mechanism must be precisely described; there is no such description in this document.

5.3.1 p 15 ln 30-35: Says pre-configured keys are commissioned. States that KEC can be used to obtain network key but this is not possible in current implementation as a cluster requires the network key for communication. Note SE 1.0 networks never transmit keys in the clear.

5.3.3 pg 15 ln 30: States that the Trust Center will pick the network key but does not add any text about entropy or randomness of the network key.

5.4.1 pg 18 ln 30: States permit joining is turned on but does not refer to or define mechanism. This is basic pre-authorization mechanism to prevent network access at the MAC layer.

5.4.1 pg 18 ln 32-35: Key derivation is a simple fixed hash and has no salting and therefore if the installation code is known the network key can be found out. Weak installation codes will also allow brute force cracking to obtain the network key.

5.4.1 pg 18 ln 37-39: The weaker link key based on the installation code is replaced by the stronger link key derived from CBKE using the KEC. Replacement of the link key is not good from a key separation standpoint and the initial link key must be retained should the device need to join again. Key domains overlap between network access and application layer.

5.4.1 pg 18 ln 41-43: The option of using KEC later initiated by the Trust Center (as opposed to the joiner as part of network access) is referred to but not detailed. Again, indicates overlapping key domains. Has implications if not strictly defined as some implementations limit the lifetime phases of certain code due to space restrictions and may have swapped out KEC code.

5.4.2.1 pg 19 ln 9: Not requiring permit join gives implicit pre-authorization to nodes with retained parameters.

5.4.2.1 pg 19 ln 18-20: Allowing rejoin at any time is mitigated by insisting the KEC-established key is used for subsequent network access (again overlapping key domains) and that the installation code can only ever be used for initial join. This may have implications for sleepy devices which have missed a network key update and then have to be re-commissioned to get back on the network, i.e. go through the registration process again to get permit join turned back on.

5.4.2.2 pg 19 ln 33-45: Authorization for rejoin is based initially on the device having network access alone. If the device does not have network access (no longer has the correct network key), authorization is then based on retention of CBKE-derived TC link key. This overlaps key domains and does not distinguish network access authorization from application layer authorization. This may be acceptable in a single ESI/single network configuration.

5.4.2.2 pg 19 ln 38-41: Sending an APS remove device may not remove the device from the network as the device will still have the network key. Selective update of network key to individual devices will be needed. This is not adequately detailed.

5.4.3: pg 20 ln 4: It is optimistic to assume a leave will be acted upon by a compromised node. At best, a leave command can only be considered a polite request, which well-behaved nodes will act on.

5.4.4 pg 20 ln 14-16: Broadcasting a new network key does not help in the case of an insider compromise. The compromised insider will get the new network key. It would be safer to allow an selective update mechanism.

5.4.4 pg 20 ln 19: It is not clear whether sleepy devices will fail to get the network key or not. This depends on the periodicity of updates. This is not specified in this document as it does depend on the mix of devices in the network. A general rule of thumb would be to rotate the network key around the same frequency as the sleepiest device on the network wakes up and polls.

5.4.5 pg 20 ln 36-37: No detail on specifying policy for link key update. Again, considered out of scope for this level of specification and depends on individual devices.

5.4.5 pg 20 ln39 - pg 21 ln 2: Issue with overlapping key domains here. The application link key (with the TC) is also the same as the network access link key. Notion of a 'stale' key, which is a valid TC link key but one which can't be used for application security, only APS commands. So link key update is again achieved using the stale key.

5.4.6 pg 23 Table 5.10: The table shows static implicit authentication levels required for the various clusters (functions). It is an arbitrary fixed list of authorization and there is no flexibility to change these due to security policy. There have been interoperability issues where clusters which only require being secured with the Network Key are also being secured with the Application Link Key. Some devices have rejected these. It is not clear in the specification that this is allowed, i.e. that application link key has implied higher security level than network key and is therefore allowed

5.4.7.1 pg 24 ln 10-11: Overlapping key domain. TC link key is strictly not Application Link Key. It is not clear what applies in peer-to-peer cases.

5.4.7.2 pg 24 ln 21-25: Allowing network only communication is weak, as key is delivered based on potentially weak installation code. Not only can an intruder cause problems for the network layer functionality, it is also allowed to perform limited application layer functions.

5.4.7.3 pg24 ln 28 – pg 25 ln 2: Overlapping key domains: Statefulness required based on link keys and there is a transition where the TC link key becomes an application link key.

5.4.7.3 pg 25 ln 7 - 10: The text states that a device will be unable to communicate with other devices; not true. Table 5.10 shows this can happen.

5.4.7.4 pg 25 ln 11 – pg 26 ln 34: This section describes how two application peers can broker a shared application link key so they can communicate peer-to-peer.

5.4.7.4 pg 25 ln 25-28: It says both nodes are required to request a link key but the diagram does not show that. It shows one node requesting and the ESP (TC) delivering to both nodes . The diagram is how the APSME-REQUEST-KEY. req primitive works in ZigBee PRO. It also says ESP but would typically have to be the TC as this is the only node devices typically have a current link key with.

5.4.7.4 p26 ln 24-30: Mixing up ESP and TC. Should refer to the TC only.

5.4.8.1 pg 27 ln 1-4: The Installation Code is subsequently hashed (without any salt) and used as a TC link key to transport the network key. The fact that the installation code is printed seriously limits its confidentiality.

5.4.8.1.1 pg 27 ln 36: Use of 48 bit installation codes is very weak to protect a 128-bit key.

5.4.8.1.2 pg 37 ln 16 – pg 57 ln 45: The Matyas-Meyer-Oseas (MMO) cryptographic hash is not a NIST-approved cryptographic hash (e.g. SHA-1, SHA256). It was chosen because it uses AES-128 block cipher, which is usually available in hardware on 802.15.4 devices due to its use for frame security.

5.4.8.2.1 pg 56 ln 7-19: Describes best practices for the ESI keeping track of registered devices (i.e. authorized devices). This is a best practice as it is strictly not part of the communications protocol.

5.4.8.2.2 pg 56 ln 27-34: Does not dictate rules for permit join and leaves it to 'business processes'. In practice, it has been found that it is left on permanently. This causes significant issues for steering to the right network and subsequent network access, especially considering weak installation codes. Also this procedure mixes network access and registration. For a single TC/single network model like SE 1.0, this is acceptable as the two can be coalesced.

5.4.8.2.3 pg 56 ln 40 – pg 57 ln 16: The operation implies that only the client (joiner) can initiate re-establishment of TC link key due to a forced removal by the TC and detecting a communications failure. This may conflict with 5.4.1 pg 18 ln 41-43, which is not clear about who initiates the update.

5.4.8.2.4 pg 57 ln 19-34: The operation described may not be sufficient to remove a malicious device with a network key who would refuse what are essentially polite requests to leave.

5.5.1 pg 58 ln 24-32: The joining process here (network access) suggests that the user may have control over permit join. In the rest of the document, network access is tied to registration so it would be consistent to state that here.

5.5.2 pg 58 ln 35 – 5.5.3 pg 59 ln 13: The commissioning modes described here are somewhat orthogonal to best practices described elsewhere in the document which tie commissioning firmly to a combined network access and registration mechanism based on out-of-band communication with the utility and an install code. It therefore seems superfluous to mention the other modes which are unlikely to be used in practice and could lead to poor practices such as permit join being left on on devices which have no UI and cannot support a push button method of allowing permit join.

5.5.4.1 pg 59 ln 23 – pg 60 ln 5: The concept of a NAN has only been loosely referred to in the remainder of the document. There is not enough detail regarding network access and registration for this configuration which involves a number of parties sharing the same network. Combined with the weakness of the installation code mechanism of joining and the general aversion to sharing a common network (unless it is considered never even remotely 'owned' by the consumer), it is recommended that the NAN configuration not be used.

5.5.4.1 pg 59 ln 35-38: It should not be possible to pre-install the network key (see comment for 5.3.1 pg 14 ln 7-9). This is even more relevant for a NAN.

5.5.4.2 pg 60 ln 16: Describes the use of a 'handheld tool'. No mechanisms have been described as to how that may be used and there are security implications for the use of a local tool which can force the network into certain modes of operation which need to be considered.

5.5.4.2 pg 60 ln 22-24: The device joining and registering may not have any UI at all, so it may be sufficient to simply power on the device for it to join and register.

5.5.4.2 pg 60 ln 27-29: Use of the word 'indicator' suggests something local to the device joining. The device joining and registering may not have any UI at all, so this may require confirmation with the utility via the backhaul network.

5.6 pg 60 ln 35 – pg 61 ln 4: It is suggested (though not mandated) that public pricing information is rebroadcast using the Inter-PAN mechanism. The Inter-PAN mechanism is completely insecure so whilst this information may not be confidential, any device receiving this information could not be sure it is authentic. An attacker could easily spoof bogus public pricing messages. Therefore it is recommended this is not used.

5.7.4 pg 61 ln 42: It says all link key updates shall use the KEC but it does not state how it is initiated or who by. This may have implications if a device has to implement server functionality for KEC. See also comment for 5.4.1 pg 18 ln 41-43.

Annex B pg 81 l 1 – pg 94 ln 45: Inter-PAN mechanism is totally insecure and should not be used.

Annex C.2 pg 95 ln 32 – pg 99 ln 42: Describes the theory behind CBKE and how it generates symmetric keys without transporting them.

Annex C.2.3 pg 96 ln 8-18: States that confidentiality and integrity of the data is provided by AES-CCM (using AES-128 bit block cipher).

Annex C.2.3 pg 96 ln 20: States that key establishment through key agreement is used to establish the symmetric key required for frame security.

Annex C.2.3 pg 96 ln 22: Should say key transport will not be considered further in this section, as it is used for the network key.

Annex C.2.5 pg 97 ln 11-30: Public key key establishment is a key agreement protocol based on a standard challenge-response.

Annex C.2.6.2 pg 99 ln 8-13: The key agreement scheme used is ECMQV. This is a standard key agreement scheme detailed in NIST SP 800-56A.

Annex C.2.6.3 pg 99 ln 28-35: It states the MAC scheme specified in SEC 1 (HMAC) is used. However, it is not one of the recommended instantiations of HMAC as it is based in the MMO cryptographic hash, which is not NIST-approved.

Annex C.3 pg 100 ln 1 – pg 114 ln 45: The KEC provides a mechanism to perform mutual authentication and key establishment for the purposes of obtaining a symmetric key for frame protection at the APS (application support) layer. This protection is in addition to the hop-by-hop frame protection at the network layer using the network key.

Annex C.3.1.2.2.1.1: pg 103 ln 9: The KEC allows other cipher suites to be specified in the future, up to a maximum of 16 in total.

Annex C.4.1 pg 115 ln 8-11: It states that the temporary link key is created by hashing a random number (i.e. installation code) with the device identifier. However, it is described differently in 5.4.8.1.2.

Annex C.4.1 pg 115 ln 22-25: Replacement of TC link key with CBKE link key is reasonable but its subsequent use as an application layer link key overlaps key domains.

Annex C.4.2 pg 115 ln 28 – pg 123 ln 17: Certificate-based Key Establishment is used to provide mutual authentication based on certificates in a PKI and key establishment.

Annex C.4.2 pg 115 ln 33-38: The information does not specify what sort of device it is or any sort of security level. It merely mentions profile specific information. This is therefore left up to the certificate issuer. It is known that the certificate issuer does not include any information about device type or security level due to the associate business process of bulk issuing certificates. This has implications for devices being to masquerade as other devices. It will be much easier to crack a simple in-home display than a meter and if this can then masquerade as a bogus meter, then it could be a serious issue.

Annex C.4.2 pg 115 ln 43 – pg 116 ln 6: The PKI contains one link to the root CA.

Annex C.4.2 pg 116 ln 10-22: The security properties of CBKE are listed.

Annex C.4.2.2.1 pg 117 ln 6-10: Curve ansit163k1 only has 80 bits of security effectively and therefore does not have the lifetime required in the TRD for SE 2.0. Therefore SE 1.0 has a limited lifetime in the market with respect to NIST SP 800-57.

Annex C.4.2.2.2 pg 117 ln 11-14: The compressed elliptic curve point representation is used.

Annex C.4.2.2.4 pg 117 ln 25 – pg 118 ln 7: The implicit certificate mechanism specifies ProfileAttributeData but is not explicit about the contents anywhere in the document

Annex C.4.2.2.4 pg 117 ln 34: SEC 4 describes the ECQV implicit certificate generation and validation. This is not NIST-approved.

Annex C.4.2.2.6 pg 118 ln 14-25: The Matyas-Meyer-Oseas (MMO) cryptographic hash is not a NIST-approved cryptographic hash. See also comment for 5.4.8.1.2 pg 37 ln 16 – pg 57 ln 45.

Annex C.4.2.3.2 pg 121 ln 5-21: The certificate validation is only described as part of the PKI checking. There is no language elsewhere in the document regarding further certificate checking based on what may be in the profile attribute data.

Annex F pg 217 ln 23-25: The network key is not sent in the clear. It is protected by the initial TC link key, based on the installation code.

Annex F pg 217 ln 36-43: Overlapping of key domains again. Initial keying material should remain distinct from subsequent TC link key, which should be different from Application link key.

Annex F pg 217 ln 41: Annex E would appear to be the wrong reference. It is also not clear that the hashing with the long address ever takes place, therefore this distinction between pre-configured link keys and temporary link keys becomes moot.

Annex F pg 218 ln 23-37: The outlining is inconsistent, causing confusion. The three bullets between these lines should be at the same outline level.

Annex F pg 219 ln 19-20: It states the message is identical if directly joining to the TC, however the APS Header and APS Auxiliary Header are clearly not obtained from the encapsulated tunnel message and are created at the TC.

Annex F pg 219 ln 32-35: The frame is not secured at the network layer but is secured at the APS layer thus the network key cannot be directly read by an eavesdropper.

Security synopsis

The following is a list of security statements with regard to ZigBee SE 1.0, based on the analysis

ID	Statement	Comment
ZSE1_0.SEC.1	Mandatory cryptographic protection of network layer frames using AES-128-CCM* to provide encryption and 32-bit message integrity code (MIC) based on shared network key	
ZSE1_0.SEC.2	Optional cryptographic protection of APS layer frames using AES-128-CCM* to provide encryption and 32-bit message integrity code (MIC) based on pairwise link key	
ZSE1_0.SEC.3	Optional use of unprotected frames using the Inter-PAN communications mechanism	
ZSE1_0.SEC.4	Network access initially based on permit join flag	Use of permit join flag based on policy
ZSE1_0.SEC.5	Network access subsequent authorization based on installation code white list	Installation code white list based on policy
ZSE1_0.SEC.6	Initial pair-wise link key between device and Trust Center is derived solely from the installation code	Entropy and randomness of installation code have a direct bearing on the strength of the KEK
ZSE1_0.SEC.7	Network key is delivered secured using key derived directly from initial TC link key	
ZSE1_0.SEC.8	Network key is updated by broadcasting new network key encrypted with key derived directly from existing network key	
ZSE1_0.SEC.9	Periodicity of network key and pair-wise link key updates is not specified	
ZSE1_0.SEC.10	Long term pairwise link key between device and Trust Center is used for application layer security	
ZSE1_0.SEC.11	Additional pair-wise link keys between peer devices are brokered through the Trust Center	
ZSE1_0.SEC.12	Authorization is static policy based on basic authentication level and function type	
ZSE1_0.SEC.13	Public key cryptography used in Key	

ID	Statement	Comment
	Establishment Cluster (KEC) to provide long term pairwise link key between device and Trust Center	
ZSE1_0.SEC.14	KEC key agreement uses ECMQV algorithm and binary elliptic curve ansit163k1, providing an equivalent symmetric key strength of 80 bits	
ZSE1_0.SEC.15	Implicit certificates are installed in devices with a single link PKI. The certificate content is not explicitly specified	
ZSE1_0.SEC.16	Public key reconstruction and implied validation from implicit certificates uses ECQV mechanism, using binary elliptic curve ansit163k1, providing an equivalent symmetric key strength of 80 bits	
ZSE1_0.SEC.17	Mutual authentication of devices is based on ECQV certificate validation	
ZSE1_0.SEC.18	The cryptographic hash algorithm uses is the Matyas-Meyer-Oseas scheme (MMO), using AES-128 as the underlying block cipher	
ZSE1_0.SEC.19	The keyed hash message authentication code used is HMAC, based on the MMO cryptographic hash algorithm	

NIST 7628 template to compare with SE 1.0

A table will be applied to the security statements. The table lists the relevant requirements in the NISTIR 7628 document in relation to a HAN communication protocol. For each security statement, the requirement should be first marked as relevant (Rel), and if relevant, whether the requirement was met or not (Met).

ZSE1_0.SEC.1

Mandatory cryptographic protection of network layer frames using AES-128-CCM* to provide encryption and 32-bit message integrity code (MIC) based on shared network key

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel														Y			
Met														Y			
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel									Y						Y	Y	Y
Met									Y						Y	Y	Y
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel		Y				Y											
Met		Y				Y											

ZSE1_0.SEC.2

Optional cryptographic protection of APS layer frames using AES-128-CCM* to provide encryption and 32-bit message integrity code (MIC) based on pairwise link key

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel														Y			
Met														Y			
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel									Y						Y	Y	Y
Met									Y						Y	Y	Y
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel		Y				Y											
Met		Y				Y											

ZSE1_0.SEC.3

Optional use of unprotected frames using the Inter-PAN communications mechanism

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel														Y			
Met														N			
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel									Y						Y	Y	Y
Met									N						N	N	N
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel		Y				Y											
Met		N				N											

Comment

- Not met due to no protection placed on inter-PAN frames

ZSE1_0.SEC.4

Network access initially based on permit join flag

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel														Y	Y		
Met														Y	Y		
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel												Y					
Met												Y					
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel																	
Met																	

ZSE1_0.SEC.5

Network access subsequent authorization based on installation code white list

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel	Y	Y	Y										Y	Y	Y		
Met	Y	Y	Y										Y	Y	Y		
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel		Y	Y	Y	Y	Y	Y										
Met		Y	Y	Y	Y	Y	Y										
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel																	
Met																	

ZSE1_0.SEC.6

Initial pair-wise link key between device and Trust Center is derived solely from the installation code

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel	Y	Y	Y										Y	Y	Y		
Met	P	P	P										P	P	P		
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel															Y	Y	Y
Met															P	P	P
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel						Y											
Met						P											

Comment

- Only partially met due to potential weakness of installation code based on entropy, randomness and potential lack of confidentiality

ZSE1_0.SEC.7

Network key is delivered secured using key derived directly from initial TC link key

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel															Y	Y	Y
Met															P	P	P
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel			Y			Y											
Met			P			P											

Comment

- Only partially met due to potential weakness of installation code based on entropy, randomness and potential lack of confidentiality

ZSE1_0.SEC.8

Network key is updated by broadcasting new network key encrypted with key derived directly from existing network key

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel															Y	Y	Y
Met															Y	Y	Y
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel			Y			Y											
Met			P			Y											

Comment

- Only partially met for SG.SC-14 due to compromised insider will still get new network key

ZSE1_0.SEC.9

Periodicity of network key and pair-wise link key updates is not specified

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel																	
Met																	
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel			Y														
Met			N														

Comment: Not met for SG.SC-14 as updates (transmission of security parameters) are not specified

ZSE1_0.SEC.10

Long term pairwise link key between device and Trust Center is used for application layer security

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel									Y	Y							
Met									N	N							
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel											Y						
Met											N						

Comment: Not met as network access is not distinguished from application layer security and key domains overlap

ZSE1_0.SEC.11

Additional pair-wise link keys between peer devices are brokered through the Trust Center

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel																	
Met																	
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel			Y														
Met			Y														

ZSE1_0.SEC.12

Authorization is static policy based on basic authentication level and function type

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel	Y	Y	Y														
Met	Y	Y	Y														
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel		Y	Y	Y													
Met		Y	Y	Y													
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel											Y						
Met											Y						

ZSE1_0.SEC.13

Public key cryptography used in Key Establishment Cluster (KEC) to provide long term pairwise link key between device and Trust Center

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel															Y	Y	
Met															Y	Y	
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel	Y	Y	Y														
Met	Y	P	Y														

ZSE1_0.SEC.14

Public key cryptography used in Key Establishment Cluster (KEC) to provide long term pairwise link key between device and Trust Center

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel															Y	Y	
Met															Y	Y	
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel	Y	Y	Y														
Met	Y	P	Y														

Comment

- Only partially for SG.SC-12 as key lifetime is short according to SP 800-57.

ZSE1_0.SEC.15

KEC key agreement uses ECMQV algorithm and binary elliptic curve ansit163k1, providing an equivalent symmetric key strength of 80 bits

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel															Y	Y	
Met															Y	Y	
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel	Y	Y	Y														
Met	Y	P	Y														

Comment

- Only partially for SG.SC-12 as key lifetime is short according to SP 800-57.

ZSE1_0.SEC.16

Implicit certificates are installed in devices with a single link PKI. The certificate content is not explicitly specified

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel		Y	Y	Y		Y	Y										
Met		Y	Y	Y		P	Y										
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel	Y	Y	Y	Y	Y												
Met	Y	P	Y	Y	P												

Comment

- Only partially met for SG.IA-5 as certificate does not explicitly specify device type
- Only partially met for SG.SC-12 as key lifetime is short according to SP 800-57
- Only partially met for SG.SC-19 as certificate does not explicitly specify roles

ZSE1_0.SEC.17

Mutual authentication of devices is based on ECQV certificate validation

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel		Y	Y	Y		Y	Y										
Met		Y	Y	Y		P	Y										
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel	Y	Y	Y	Y	Y												
Met	Y	N	Y	Y	P												

Comment

- Only partially met for SG.IA-5 as certificate does not explicitly specify device type
- Not met for SG.SC-12 as ECQV is not NIST-approved
- Only partially met for SG.SC-19 as certificate does not explicitly specify roles

ZSE1_0.SEC.18

The cryptographic hash algorithm uses is the Matyas-Meyer-Oseas scheme (MMO), using AES-128 as the underlying block cipher

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel																	
Met																	
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel		Y															
Met		N															

Comment

- Not met for SG.SC-12 as MMO is not NIST-approved

ZSE1_0.SEC.19

The keyed hash message authentication code used is HMAC, based on the MMO cryptographic hash algorithm

Req.	SG.AC.2	SG.AC-3	SG.AC-4	SG.AC-5	SG.AC-6	SG.AC-7	SG.AC-8	SG.AC-10	SG.AC-11	SG.AC-12	SG.AC-13	SG.AC-14	SG.AC-15	SC.AC-16	SG.AC-17	SG.AC-20	SG.AC-21
Rel																	
Met																	
Req.	SG.AU-2	SG.AU-3	SG.AU-4	SG.AU-7	SG.AU-8	SG.AU-9	SG.AU-13	SG.AU-14	SG.AU-15	SG.AU-16	SG.CM-1	SG.CM-3	SG.CM-4	SG.CM-5	SG.CM-6	SG.CM-7	SG.CM-10
Rel																	
Met																	
Req.	SG.CM-11	SG.IA-1	SG.IA-2	SG.IA-3	SG.IA-4	SG.IA-5	SG.IA-6	SG.SC-1	SG.SC-2	SG.SC-3	SG.SC-4	SG.SC-5	SG.SC-6	SG.SC-7	SG.SC-8	SG.SC-9	SG.SC-10
Rel																	
Met																	
Req.	SG.SC-11	SG.SC-12	SG.SC-14	SG.SC-15	SG.SC-19	SG.SC-20	SG.SC-21	SG.SC-23	SG.SC-25	SG.SC-26	SG.SC-29	SG.SC-30	SG.SI-8	SG.SI-9			
Rel		Y															
Met		P															

Comment

- Only partially met for SG.SC-12 as while HMAC is NIST-approved, the underlying cryptographic hash (MMO) is not NIST-approved